

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computer-readable medium having a base generator class stored thereon for use by developers to create generators to perform specific tasks, the base generator class comprising:

a base generator class constructor;

a generator properties class that provides incrementation capability, which allows the value of a generator property to vary during consecutive executions of a generator;

a status indicator;

a schedule class; [[and]]

a logging class; and

wherein the logging class is used to verify the tasks performed by the generators.

2. (Previously amended) The computer-readable medium of Claim 1, wherein the generator properties class that provides incrementation capability includes a plurality of generator properties.

3. (Original) The computer-readable medium of Claim 2, wherein said plurality of generator properties includes:

a value of a generator property;

a plurality of incrementation settings;

a default incrementor that changes the value of the generator property; and

a default validator that validates the value of the generator property.

4. (Original) The computer-readable medium of Claim 1, wherein the status indicator includes a status user interface (UI) for displaying the execution status of generators.

5. (Previously presented) The computer-readable medium of Claim 1, wherein the schedule class comprises:

- a start condition under which the execution of a generator may be started;
- a recurrence condition under which the execution of a generator may recur;
- an end condition under which the execution of a generator stops; and
- a dialog box that can be used to accept user input.

6. (Previously presented) The computer-readable medium of Claim 1, wherein the logging class enables the recording of the execution process of a generator.

7. (Currently amended) A method of creating a generator, wherein the generator performs a specific task ~~such as~~ including at least creating a file, comprising:

- creating a new generator class that inherits a base generator class that contains incrementation capability;

- creating a public default constructor for the new generator class that overrides the base generator class constructor; [[and]]

- implementing a function in the new generator class to perform the specific task; and
verifying the task based on properties of the generator.

8. (Original) The method of Claim 7, wherein creating a public default constructor comprises:

- initializing the base generator class constructor with the name and the description of the generator; and

- defining the properties of the generator.

9. (Original) The method of Claim 8, wherein defining properties for the generator comprises:

- (a) defining the name of a property;
- (b) setting a default value for the property;
- (c) providing a description for the property;
- (d) specifying incrementation settings for the property;
- (e) creating a custom property incrementor, if applicable;
- (f) creating a custom property validator, if applicable; and
- (g) repeating (a)-(f) for all properties of the generator.

10. (Original) The method of Claim 7, further comprising implementing a function to be executed before each execution of a generator.

11. (Original) The method of Claim 7, further comprising implementing a function to be executed after each execution of a generator.

12. (Currently amended) A method of using a generator that performs a specific task ~~such as~~ including at least creating a file, comprising:

customizing settings of ~~[[a]]~~ the generator, the settings ~~include~~ including incrementation settings that specify how the value of a generator property may vary between generated objects; ~~[[and]]~~

executing the generator with the customized settings; and
verifying the task based on the settings of the generator.

13. (Currently amended) The method of Claim 12, customizing the settings of ~~[[a]]~~ the generator, is accomplished through a user interface.

14. (Currently amended) The method of Claim 13, further comprising:
starting an object generator user interface;
selecting ~~[[a]]~~ the generator; and
customizing properties of the generator;
15. (Currently amended) The method of Claim 14, selecting a generator further comprising adding ~~[[a]]~~ the generator from files containing one or more generators.
16. (Original) The method of Claim 14, further comprising loading the settings of a generator from a file.
17. (Currently amended) The method of Claim 14, wherein customizing the properties of the generator comprises:
- (a) selecting ~~[[a]]~~ one property;
 - (b) specifying ~~[[the]]~~ a value of the one property;
 - (c) specifying incrementation settings of the one property; and
 - (d) repeating (a)-(c) until there are no more properties to be customized.
18. (Original) The method of Claim 14, further comprising setting a schedule for executing the generator.
19. (Original) The method of Claim 14, further comprising setting logging options for executing the generator.
20. (Original) The method of Claim 14, further comprising saving the settings of the generator.

21. (Currently amended) The method of Claim 12, customizing the settings of a generator[[,]] is accomplished programmatically.

22. (Currently amended) The method of Claim 21, further comprising:
creating a new instance of the generator;
setting [[the]] a number of objects to be generated by the generator; and
customizing [[the]] properties of the generator.

23. (Currently amended) The method of Claim 22, wherein customizing [[the]] properties of the generator comprises:

- (a) setting ~~the value of a property;~~ values of the properties; and
- (b) specifying incrementation settings of the properties ~~property;~~ and
- (c) ~~repeating (a) (b) until there are no more properties to be customized.~~

24. (Original) The method of Claim 21, further comprising:
creating a new instance of the generator; and
loading saved settings of the generator from a file.

25. (Original) The method of Claim 21, further comprising:
creating a new instance of the generator;
loading saved settings of the generator from a file; and
implementing a function to execute the generator asynchronously.

26. (Original) The method of Claim 21, further comprising:
creating a new instance of the generator;
loading saved settings of the generator from a file;
displaying an object generation status UI; and

displaying an object generation status UI; and
adding the generator to the object generation status UI.

27. (Original) The method of Claim 21, further comprising:
creating a new instance of the generator;
loading saved settings of the generator from a file;
displaying a schedule dialog box that allows a user to specify a schedule for executing the generator; and
displaying a logging dialog box that allows a user to specify logging options for executing the generator.

28. (Original) The method of Claim 12, further comprising executing the generator through a user interface.

29. (Original) The method of Claim 12, further comprising executing the generator programmatically.

30. (Currently amended) A method for object generation using a base generator class, comprising:

creating a generator that performs a specific task;
customizing settings of the generator, the settings ~~include~~ including incrementation settings that specify how the value of a generator property may vary between generated objects;
[[and]]
executing the generator with the customized settings; and
verifying the task based on the settings of the generator.

31. (Original) The method of Claim 30, wherein creating a generator that performs a specific task comprises:

creating a new generator class that inherits the base generator class;
creating a public default constructor for the new generator class that overrides the base generator class constructor; and
implementing a function in the new generator class to perform a specific task.

32. (Original) The method of Claim 31, creating a public default constructor further comprising:

initializing the base generator class constructor with the name and the description of the generator; and
defining the properties of the generator.

33. (Currently amended) The method of Claim 32, wherein defining the properties for the generator includes[[]]:

- (a) defining ~~the name of a property~~ names of the properties;
- (b) setting ~~a default value for the property~~ default values for the properties;
- (c) providing ~~a description for the property~~ descriptions for the properties;
- (d) specifying incrementation settings for the ~~property~~ properties;
- (e) creating a custom property incrementor, if applicable; and
- (f) creating a custom property validator, if applicable; ~~and~~
- ~~(g) repeating (a) (f) for all properties of the generator.~~

34. (Currently amended) The method of Claim 31, further comprising implementing a function to be executed before each execution of [[a]] the generator.

35. (Currently amended) The method of Claim 31, further comprising implementing a function to be executed after each execution of [[a]] the generator.

36. (Original) The method of Claim 30, customizing the settings of a generator is accomplished through a user interface.

37. (Currently amended) The method of Claim 36, further comprising:
starting an object generator user interface;
selecting [[a]] the generator; and
customizing [[the]] properties of the generator[[:]].

38. (Currently amended) The method of Claim 37, wherein selecting [[a]] the generator comprises adding [[a]] the generator from files containing one or more generators.

39. (Original) The method of Claim 37, further comprising loading generator settings from a file.

40. (Currently amended) The method of Claim 37, wherein customizing the properties of the generator comprises:

- (a) selecting [[a]] one property;
- (b) specifying the value of the one property;
- (c) specifying incrementation settings of the one property; and
- (d) repeating (a)-(c) until there are no more properties to be customized.

41. (Original) The method of Claim 37, further comprising setting a schedule for executing the generator.

42. (Original) The method of Claim 37, further comprising setting logging options for executing the generator.

43. (Original) The method of Claim 37, further comprising saving the settings of the generator.

44. (Previously presented) The method of Claim 30, customizing the settings of a generator is accomplished programmatically.

45. (Currently amended) The method of Claim 44, further comprising:
creating a new instance of the generator;
setting [[the]] a number of objects to be generated by the generator; and
customizing the properties of the generator.

46. (Currently amended) The method of Claim 45, wherein customizing the properties of the generator comprises:

- (a) setting [[the]] a value of [[a]] one property;
- (b) specifying incrementation settings of the one property; and
- (c) repeating (a)-(b) until there are no more properties to be customized.

47. (Original) The method of Claim 44, further comprising:
creating a new instance of the generator; and
loading saved settings of the generator from a file.

48. (Original) The method of Claim 44, further comprising:
creating a new instance of the generator;
loading saved settings of the generator from a file; and

implementing a method to execute the generator asynchronously.

49. (Original) The method of Claim 44, further comprising:

creating a new instance of the generator;

loading saved settings of the generator from a file;

displaying an object generation status UI; and

adding the current generator to the object generation status UI.

50. (Original) The method of Claim 44, further comprising:

creating a new instance of the generator;

loading saved settings of the generator from a file;

displaying a schedule dialog box that allows a user to specify a schedule for executing the generator; and

displaying a logging dialog box that allows a user to specify logging options for executing the generator.

51. (Original) The method of Claim 30, further comprising executing the generator through a user interface.

52. (Original) The method of Claim 30, further comprising executing the generator programmatically.

53. (Currently amended) A method of varying ~~[[the]]~~ a value of a property associated with a task, during consecutive executions of the task, comprising:

allowing the value of the property to vary during consecutive executions of the task;

creating settings associated with the property that control how the value may vary during consecutive executions of the task; ~~[[and]]~~

allowing a user executing the task to customize the settings according to user preference;
and
verifying the task based on the settings.

54. (Original) The method of Claim 53, wherein the step of allowing the value of the property to vary during consecutive executions of the task further comprises:

implementing a function that increments a property value according to the settings associated with the property that control how the value may vary during consecutive executions of the task.

55. (Currently amended) A computer-readable medium containing computer-executable instructions for a method of varying [[the]] a value of a property associated with a task, during consecutive executions of the task, the method comprising:

allowing the value of the property to vary during consecutive executions of the task;

creating settings associated with the property that control how the value may vary during consecutive executions of the task; [[and]]

allowing a user executing the task to customize the settings according to user preference;
and
verifying the task based on the settings.

56. (Original) The computer-readable medium of Claim 55, wherein the step of allowing the value of the property to vary during consecutive executions of the task further comprises:

implementing a function that increments a property value according to the settings associated with the property that control how the value may vary during consecutive executions of the task.